

Binary Quantum Search

Vladimir E. Korepin*

*C.N. Yang Institute for Theoretical Physics,
State University of New York at Stony Brook,
Stony Brook, NY 11794-3840*

Ying Xu[†]

*Department of Physics and Astronomy,
State University of New York at Stony Brook,
Stony Brook, NY 11794-3800*

(Dated: February 5, 2008)

Abstract

Database search has wide applications and is used as a subroutine in many important algorithms. We shall consider a database with one target item. Quantum algorithm finds the target item in a database faster than any classical algorithm. It frequently occurs in practice that only a portion of information about the target item is interesting, or we need to find a group of items sharing some common feature as the target item. This problem is in general formulated as search for a part of the database [a block] containing the target item, instead of the item itself. This is partial search. Partial search trades accuracy for speed, i.e. it works faster than a full search. Partial search algorithm was discovered by Grover and Radhakrishnan. We shall consider optimized version of the algorithm and call it GRK. It can be applied successively [in a sequence]. First the database is partitioned into blocks and we use GRK to find the target block. Then this target block is partitioned into sub-blocks and we use GRK again to find the target sub-block. [We can call it binary quantum search.] Another possibility is to partition the database into sub-blocks directly and use GRK to find the target sub-block in one time. In this paper we prove that the latter is faster [makes less queries to the oracle].

PACS numbers: 03.67.-a, 03.67.Lx

*Electronic address: korepin@insti.physics.sunysb.edu

†Electronic address: yixu@ic.sunysb.edu

I. INTRODUCTION

Database search has many applications. Search algorithm enters as a subroutine in many important algorithms [1, 2, 3] in computer sciences. Grover discovered a quantum algorithm [4] which searches a database faster than any classical algorithm. Let's consider a database with one target item. We use number of queries to the oracle as complexity measure. The Grover algorithm finds the target item [with probability 1] in

$$j_{\text{full}} = \frac{\pi}{4}\sqrt{N}, \quad N \rightarrow \infty \quad (1)$$

iterations [queries to the oracle]. We shall call it a full search.

It occurs frequently in practice that less information is needed. For example, the address of the target item in binary form is $|t\rangle = |b_1b_2b_3\dots b_n\rangle$, and we want to find only the first 3 bits $b_1b_2b_3$. This means that the database is partitioned into 8 blocks. All items in a block share the common feature such that the first 3 bits being the same. We want to find the block containing the target item. This is an example of partial search. The general problem of partial search considers the following: An N item database is partitioned into K blocks, each of the same size

$$b = \frac{N}{K}. \quad (2)$$

A user wants to find the block containing the target item, instead of the target item itself. The block with the target item is called the target block; others non-target blocks. Partial search naturally arises in list matching [5]. Partial search is not only a compromise on accuracy for speed, but also has its own significance. The GRK algorithm of partial search was suggested by Grover and Radhakrishnan [6], and optimized in [7]. It takes $\sim \frac{\pi}{4}(1 - \text{coeff}(K))\sqrt{N}$ number of queries to find the target block. Here $\text{coeff}(K)$ is a finite positive number, which depends on K and has a limit when blocks are large $b \rightarrow \infty$. GRK is the most efficient partial search algorithm known in literatures [6, 7, 8, 9, 10].

GRK can be applied in a sequence [one after another], i.e. after the first GRK, the target block found can be further partitioned into sub-blocks. Then a second GRK can be applied to find the sub-block containing the target item [called the target sub-block]. We shall call the sequence of GRK's a partial search hierarchy. In hierarchical search we iterate GRK. A practical example would be: In order to find a hotel, we first look at a State map and then a town map. We shall see that the second GRK works faster than the first one. Actually,

GRK can be conducted repeatedly until we find the smallest target sub-sub-block interested. The total number of queries is the sum of queries of each GRK in the hierarchy. [We use number of queries as measure of complexity.]

Alternative to a partial search hierarchy which finds the target sub-sub-block, we could partition the database directly into sub-sub-blocks and use GRK once: We shall call it direct partial search. Although each GRK works faster than the previous one in the hierarchy, it is not guaranteed that the total number of queries in the hierarchy [sequence of GRK's] is less than that of a direct partial search. On the contrary, we will prove that *direct partial search works faster*, which is the main result of the paper. For example, consider a database partitioned into 2 blocks. Each block is partitioned into 2 sub-blocks, so totally 4 sub-blocks. One could first find the target block using GRK, then the target sub-block using sequential GRK. However, it is faster to run a GRK directly over the 4 sub-blocks, which finds the target sub-block once.

The paper consists of two parts: In the first part, we start with the Grover algorithm and the GRK algorithm. Then we study the partial search hierarchy in detail. The second part proceeds to a comparison of the hierarchical partial search with direct partial search. Then we prove our main result that direct partial search works faster.

II. THE GROVER SEARCH ALGORITHM

In our paper, we consider different methods of partial search. They are all built on the original idea of the full Grover search [2, 4, 11]. Let's formulate the problem. Consider a database of N items with one target item [14]. The database is associated with a Hilbert space with N normalized basis vectors. The basis vector corresponding to item x is denoted by $|x\rangle$. The Grover search is a quantum algorithm which starts from the uniform superposition of all basis vectors in the whole database:

$$|s_1\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle, \quad \langle s_1 | s_1 \rangle = 1. \quad (3)$$

The algorithm searches for a single target item $|t\rangle$ iteratively. The Grover iteration is a unitary transform:

$$G_1 = -I_{s_1} I_t. \quad (4)$$

Later we shall call it a global iteration in GRK. Here I_t and I_{s_1} are two inversions about the target item $|t\rangle$ and the uniform superposition $|s_1\rangle$ defined in (3), respectively:

$$I_t = \hat{I} - 2|t\rangle\langle t|, \quad (5)$$

$$I_{s_1} = \hat{I} - 2|s_1\rangle\langle s_1|, \quad (6)$$

where \hat{I} is the identical operator. The Grover iteration G_1 is a rotation [11] in the Hilbert space from $|s_1\rangle$ towards the target $|t\rangle$ by an angle θ_1 defined by:

$$\sin^2 \theta_1 = \frac{1}{N}. \quad (7)$$

After j_1 iterations the state of the database becomes [2, 11, 12]:

$$G_1^{j_1} |s_1\rangle = \sin((2j_1 + 1)\theta_1) |t\rangle + \frac{\cos((2j_1 + 1)\theta_1)}{\sqrt{N-1}} \sum_{x \neq t}^{N-1} |x\rangle. \quad (8)$$

Therefore after $j_{\text{full}} = \pi/(4\theta_1) - 1/2$ iterations the probability amplitude of $|t\rangle$ becomes unity and amplitudes of other items all vanish. i.e.

$$G_1^{j_{\text{full}}} |s_1\rangle = |t\rangle. \quad (9)$$

As N becomes large $j_{\text{full}} = \pi/(4\theta_1) - 1/2$ approaches (1). More details on Grover search can be found in [2].

III. ALGORITHMS FOR PARTIAL SEARCH

Before introducing the GRK partial search algorithm [see next section], it worth mentioning a few other algorithms for comparison:

a) Naive Search

Pick a block randomly and make a full Grover search in it [which makes $\frac{\pi}{4}\sqrt{\frac{N}{K}}$ queries to the oracle]. If we find the target item then we understand that this is the target block. If not, then we discard this block and pick another randomly. Make a full Grover search in it and repeat this procedure till we find the target block. In the worst case the target block will be the last one. So with probability 1 we have to use

$$r(N, K) = \frac{(K-1)}{\sqrt{K}} \frac{\pi}{4} \sqrt{N} \quad (10)$$

iterations [queries] to find the target block, see [15].

A full Grover search finds the target item in $(\pi/4)\sqrt{N}$ queries. If we know the exact address of the target item then we also know the target block. Comparing $(\pi/4)\sqrt{N}$ with $r(N, K)$ in (10), we see that the naive version is faster only for two blocks $K = 2$. [If $K \geq 3$ a full search is faster].

b) Binary Search

Assume that $K = 2^k$ with k a positive integer. Divide the database into two blocks and make a full Grover search in one block. If the target item not found, then take the remaining block and divide it into two sub-blocks. Pick a sub-block randomly and make a full search again in it. Repeat the procedure until we are left with the last block. In the worst case, the number of queries necessary to find the target block is

$$\frac{\pi}{4}\sqrt{N}\left\{\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{4}} + \dots + \frac{1}{2^{k/2}}\right\}, \quad k = \log_2 K. \quad (11)$$

The first two terms in the braces of (11) are greater than 1 for $K \geq 3$,

$$\frac{1}{\sqrt{2}} + \frac{1}{2} = \frac{\sqrt{2} + 1}{2} > 1. \quad (12)$$

So this algorithm is less efficient than a full Grover search, when $K > 2$.

c) Grover and Radhakrishnan Version

A faster version was found in [6]. Pick randomly a block and make a full Grover search in the compliment [all items in the rest of the database]. Either the target item [and block] is found after the search or the picked block is the target block. This requires $\frac{\pi}{4}\sqrt{b(K-1)} = \frac{\pi}{4}\sqrt{N}\sqrt{\frac{K-1}{K}}$ queries. It is faster than a full search.

IV. THE GRK PARTIAL SEARCH ALGORITHM

Grover and Radhakrishnan also discovered a faster quantum algorithm [6] for partial search, which uses the same oracle as the main Grover algorithm. [See Summary and Appendix D.] Partial search also starts from the uniform superposition of all basis states (3). A general structure of the algorithm is [6, 7, 8, 9, 10, 13]:

Step 1. Global iterations: j_1 standard Grover iterations (4). After this step the state of database is $G_1^{j_1}|s_1\rangle$.

Step 2. Simultaneous local iterations in each block: j_2 local Grover iterations defined in (13) below. After step 2 the state of database is $G_2^{j_2}G_1^{j_1}|s_1\rangle$.

Local iteration is defined by

$$G_2 = \bigoplus_{\text{blocks}}^K G_2^{\text{one block}} = - \left(\bigoplus_{\text{blocks}}^K I_{s_2} \right) I_t. \quad (13)$$

It is a direct sum of Grover iterations [called local queries] defined in each block

$$G_2^{\text{one block}} = -I_{s_2}I_t. \quad (14)$$

In the expression I_t is the same inversion (5), i.e. query to the oracle. I_{s_2} is a local inversion

$$I_{s_2} = \hat{I} - 2|s_2\rangle\langle s_2|. \quad (15)$$

Here $|s_2\rangle$ is the uniform superposition of items in one block

$$|s_2\rangle = \frac{1}{\sqrt{b}} \sum_{\text{one block}}^{\text{b items}} |x\rangle. \quad (16)$$

Local iteration G_2 is a the Grover iteration in each block done simultaneously in all blocks. G_2 acts trivially on non-target blocks. A non-trivial operation [rotation] is present only in the target block with new rotation angle θ_2 defined by

$$\sin^2 \theta_2 = \frac{K}{N} = \frac{1}{b}. \quad (17)$$

Note that amplitudes of all items in non-target blocks remain intact.

Step 3. Location of the target block with a final global iteration [7, 8, 10]:

We have to vanish amplitudes of all items in non-target blocks. We can do it by application of one more global iteration. The resulting state is

$$|d\rangle \equiv G_1 G_2^{j_2} G_1^{j_1} |s_1\rangle = \sin \omega |t\rangle + \frac{\cos \omega}{\sqrt{b-1}} \sum_{\substack{\text{b-1 items} \\ x \neq t \\ \text{target block}}} |x\rangle. \quad (18)$$

The final state (18) is expressed as a superposition over items in the target block only. This is realized by requiring that the amplitude of any non-target block vanishes after the partial search, i.e.

$$\langle x|d\rangle = 0. \quad (19)$$

Here x is an arbitrary item in any non-target block. This vanishing condition can be written explicitly as an equality for j_1 and j_2 , see [7]. We shall call it a cancellation condition.

This partial search algorithm was further optimized in [7]. In the large block limit $b \rightarrow \infty$, the total number of items also large $N \rightarrow \infty$, while the ratio $K = N/b$ kept finite. Then the expression for rotation angles (7) and (17) simplifies

$$\theta_1 \rightarrow \frac{1}{\sqrt{N}}, \quad \theta_2 \rightarrow \frac{1}{\sqrt{b}}. \quad (20)$$

It turns out convenient to rewrite numbers of iterations in a scale form [6]

$$j_1 = \left(\frac{\pi}{4} - \frac{\eta}{\sqrt{K}} \right) \sqrt{N}, \quad j_2 = \frac{\alpha}{\sqrt{K}} \sqrt{N}. \quad (21)$$

Here η and α are parameters of order 1 [they have a limit]. The ranges of these parameters are discussed in Appendix B. The vanishing condition (19) in terms of these parameters simplifies in the large b limit [7, 9]

$$\tan \left(\frac{2\eta}{\sqrt{K}} \right) = \frac{2\sqrt{K} \sin 2\alpha}{K - 4 \sin^2 \alpha}. \quad (22)$$

The total number of queries is

$$S(K) \equiv j_1 + j_2 + 1 \xrightarrow{b \rightarrow \infty} \left(\frac{\pi}{4} + \frac{\alpha - \eta}{\sqrt{K}} \right) \sqrt{N}. \quad (23)$$

It was minimized [subject to the constraint (22)] in [7]. The minimum number of queries is achieved at

$$\eta(K) = \frac{1}{2} \sqrt{K} \arctan \left(\frac{\sqrt{3K-4}}{K-2} \right), \quad \alpha(K) = \frac{1}{2} \arccos \left(\frac{K-2}{2(K-1)} \right). \quad (24)$$

Thus the minimized number of queries of GRK partial search [as a function of K] is

$$S(K) \xrightarrow{b \rightarrow \infty} \left(\frac{\pi}{4} + \frac{\alpha(K) - \eta(K)}{\sqrt{K}} \right) \sqrt{N}. \quad (25)$$

A proof of (24) being the minimum is given in Appendix C. Note that $\alpha - \eta$ is negative and number of blocks $K \geq 2$ in a non-trivial situation.

In the large block limit, the ω appeared in (18) is

$$\omega = \alpha(K), \quad (26)$$

see [7]. As a consequence, the state of database after GRK (18) is the following: The amplitudes of items in non-target blocks all vanish and the state of the target block is

$$|d\rangle = \sin \alpha(K)|t\rangle + \frac{\cos \alpha(K)}{\sqrt{b-1}} \sum_{\substack{x \neq t \\ \text{target block}}}^{\text{b-1 items}} |x\rangle. \quad (27)$$

V. THE PARTIAL SEARCH HIERARCHY

A partial search hierarchy is a sequence of GRK's. After location of the target block, we may consider a subsequent GRK partial search: The target block is further partitioned into \tilde{K} sub-blocks and we search for the sub-block containing the target item [target sub-block]. For example we can use Google Earth to find the State of New York first on the map of USA and then make a sequential search for Stony Brook in the State map.

We shall show below that a sequential GRK can be done faster than the first GRK. The coefficient $\pi/4$ in (25) is replaced by a smaller number:

$$\frac{\pi}{4} \rightarrow \frac{\pi}{4} - \frac{1}{4} \arccos \left(\frac{K-2}{2(K-1)} \right). \quad (28)$$

Each successive GRK works faster than the previous one for two reasons. First, the new database is smaller [only one block of the previous one]. Second, the initial state of the new database (27) can be represented in different forms (30) and (38) below. We see that for sequential GRK, the initial state is no longer a uniform superposition of basis vectors of the new database. It is an unevenly weighted superposition with emphasis on the target $|t\rangle$, see (30) and (38). In other words, the new initial state of the database is equivalent to a partially searched [though not fully searched] one. This fact was studied in [7]. It was shown that after the first GRK the state of the target block [new database] can be written as [(27) rewritten]

$$|d\rangle = G_1 G_2^{j_2} G_1^{j_1} |s_1\rangle = \sin \alpha(K)|t\rangle + \frac{\cos \alpha(K)}{\sqrt{b-1}} \sum_{\substack{x \neq t \\ \text{target block}}}^{\text{b-1 items}} |x\rangle. \quad (29)$$

We have used relation (26). Compared with (8), we see that the state after the first GRK (29) takes the form

$$|d\rangle = G_1 G_2^{j_2} G_1^{j_1} |s_1\rangle = G_2^{\frac{\alpha(K)}{2} \sqrt{b}} |s_2\rangle, \quad (30)$$

which serves as the initial state of the sequential GRK.

For notational convenience, we use a "∼" to indicate variables in sequential GRK and make the following definitions:

$$\text{Number of items in new database :} \quad \tilde{N} = b = \tilde{K}\tilde{b}, \quad (31)$$

$$\text{Uniform superposition of new database :} \quad |\tilde{s}_1\rangle = |s_2\rangle, \quad (32)$$

$$\text{New global inversion :} \quad I_{\tilde{s}_1} = I_{s_2}, \quad (33)$$

$$\text{New global iteration :} \quad \tilde{G}_1 = G_2, \quad \tilde{\theta}_1 = \theta_2, \quad (34)$$

$$\text{Uniform superposition of one sub-block :} \quad |\tilde{s}_2\rangle = \frac{1}{\sqrt{\tilde{b}}} \sum_{\substack{\text{one} \\ \text{sub-block}}}^{\tilde{b} \text{ items}} |x\rangle, \quad (35)$$

$$\text{New local inversion :} \quad I_{\tilde{s}_2} = I - 2|\tilde{s}_2\rangle\langle\tilde{s}_2|, \quad (36)$$

$$\text{New local iteration :} \quad \tilde{G}_2 = -I_{\tilde{s}_2}I_t, \quad \sin^2 \tilde{\theta}_2 = \frac{1}{\tilde{b}}. \quad (37)$$

Written in these notations, the initial state of new database (30) is equivalent to a partially searched one with $\frac{\alpha(K)}{2}\sqrt{\tilde{N}}$ new global queries, i.e.

$$|d\rangle = G_1 G_2^{j_2} G_1^{j_1} |s_1\rangle = \tilde{G}_1^{\frac{\alpha(K)}{2}\sqrt{\tilde{N}}} |\tilde{s}_1\rangle \quad (38)$$

Steps of sequential GRK can be written similarly to the first GRK using new notations (31)-(37). The resultant state of target sub-block is

$$|\tilde{d}\rangle \equiv \tilde{G}_1 \tilde{G}_2^{\tilde{j}_2} \tilde{G}_1^{\tilde{j}_1} \left(\tilde{G}_1^{\frac{\alpha(K)}{2}\sqrt{\tilde{N}}} |\tilde{s}_1\rangle \right) = \sin \tilde{\omega} |t\rangle + \frac{\cos \tilde{\omega}}{\sqrt{\tilde{b}-1}} \sum_{\substack{\text{target} \\ \text{sub-block}}}^{\tilde{b}-1 \text{ items}} |x\rangle. \quad (39)$$

Note that the vector in the parenthesis is $|d\rangle$ of (27). We also have [similar to (19)]

$$\langle x|\tilde{d}\rangle = 0, \quad \forall x \in \{\text{items of non-target sub-blocks}\}. \quad (40)$$

This yields cancellation condition relating \tilde{j}_1 and \tilde{j}_2 , see [7]. We introduce parameters $\tilde{\eta}$ and $\tilde{\alpha}$ defined by

$$\tilde{j}_1 = \left(\frac{\pi}{4} - \frac{\alpha(K)}{2} - \frac{\tilde{\eta}}{\sqrt{\tilde{K}}} \right) \sqrt{\tilde{N}}, \quad \tilde{j}_2 = \tilde{\alpha} \sqrt{\tilde{b}}. \quad (41)$$

The algorithm is also optimized [7] in the large sub-block limit: $\tilde{b} \rightarrow \infty$, $\tilde{N} \equiv \tilde{K}\tilde{b} \rightarrow \infty$. In the limit, the angles (34) and (37) simplify

$$\tilde{\theta}_1 = \frac{1}{\sqrt{\tilde{N}}}, \quad \tilde{\theta}_2 = \frac{1}{\sqrt{\tilde{b}}}. \quad (42)$$

The minimum is achieved at

$$\tilde{\eta}(\tilde{K}) = \frac{1}{2}\sqrt{\tilde{K}} \arctan\left(\frac{\sqrt{3\tilde{K}-4}}{\tilde{K}-2}\right) = \eta(\tilde{K}), \quad \tilde{\alpha}(\tilde{K}) = \frac{1}{2} \arccos\left(\frac{\tilde{K}-2}{2(\tilde{K}-1)}\right) = \alpha(\tilde{K}) \quad (43)$$

Similar to (26), we have in the large sub-block limit

$$\tilde{\omega} = \alpha(\tilde{K}). \quad (44)$$

As a result the number of queries of the sequential GRK is

$$\bar{S}(K, \tilde{K}) \equiv \tilde{j}_1 + \tilde{j}_2 + 1 \xrightarrow{\tilde{b} \rightarrow \infty} \left(\frac{\pi}{4} - \frac{\alpha(K)}{2} + \frac{\alpha(\tilde{K}) - \eta(\tilde{K})}{\sqrt{\tilde{K}}} \right) \sqrt{\tilde{N}}. \quad (45)$$

In principle, sequential GRK's can be conducted successively until the smallest target sub-sub-block is found. Here arises a question on the efficiency of hierarchical partial search, i.e. whether or not is a sequence of GRK's works faster than a direct GRK partial search of the smallest sub-sub-blocks. As will be shown in the following section, direct GRK partial search makes less queries in the quantum case.

VI. COMPARISON OF HIERARCHICAL PARTIAL SEARCH WITH DIRECT PARTIAL SEARCH

The partial search hierarchy forms a sequence of GRK's. It starts from searching for the largest target block and ends with searching for the smallest target sub-sub-block. On the other hand, it is also possible to partition the database directly into the smallest sub-sub-blocks and use a GRK to find the target sub-sub-block in one time. One question of significance is **whether the hierarchical search works faster than the direct search or not**. This question is of practical importance and the answer turns out to be *negative*. We prove the statement by studying the first two successive GRK's in the hierarchy.

We have already derived the optimized number of queries of the first two GRK's in (25) and (45), respectively. So that the total number of queries is the sum:

$$\begin{aligned} T(K, \tilde{K}) &\equiv S(K) + \bar{S}(K, \tilde{K}) \\ &= \left\{ \frac{\pi}{4} + \left[\frac{\pi}{4} + \frac{1}{2}\alpha(K) - \eta(K) \right] \frac{1}{\sqrt{K}} + \left[\alpha(\tilde{K}) - \eta(\tilde{K}) \right] \frac{1}{\sqrt{K\tilde{K}}} \right\} \sqrt{N}. \end{aligned} \quad (46)$$

On the other hand, if the database is partitioned directly into $K\tilde{K}$ blocks, a direct GRK algorithm would require

$$S(K\tilde{K}) = \left[\frac{\pi}{4} + \frac{\alpha(K\tilde{K}) - \eta(K\tilde{K})}{\sqrt{K\tilde{K}}} \right] \sqrt{N} \quad (47)$$

queries instead. Let us compare $T(K, \tilde{K})$ and $S(K\tilde{K})$, assuming that both $K \geq 2$ and $\tilde{K} \geq 2$.

A. Numerical Comparison of Query Numbers and Asymptotic Analysis

Before giving the complete proof, we illustrate this fact by looking at a few concrete examples. Here in Table I we give a few numerical examples of query numbers $S(K\tilde{K})$ and $T(K, \tilde{K})$ as well as their difference, for a better understanding. It is clear that each $T - S$ is positive in the last column.

TABLE I: Numerical Examples of Query Numbers

K	\tilde{K}	$S(K\tilde{K})/\sqrt{N}$	$T(K, \tilde{K})/\sqrt{N}$	$(T(K, \tilde{K}) - S(K\tilde{K}))/\sqrt{N}$
2	2	0.61548	0.670379	0.054899
2	3	0.646015	0.695421	0.049406
3	2	0.646015	0.721158	0.075143
2	4	0.664521	0.71289	0.048369
4	2	0.664521	0.73929	0.074769
3	3	0.671394	0.741605	0.070211

Independently, we also look at the case that number of blocks and sub-blocks both being large, i.e. $K \rightarrow \infty$, $\tilde{K} \rightarrow \infty$. Asymptotic forms of $\alpha(x)$ and $\eta(x)$ are obtained from (24) as

$$\alpha(x) \sim \frac{\pi}{6} + \frac{1}{2\sqrt{3}x} + \frac{5\sqrt{3}}{(6x)^2}, \quad \eta(x) \sim \frac{\sqrt{3}}{2} + \frac{1}{2\sqrt{3}x} + \frac{11\sqrt{3}}{90x^2}, \quad x \rightarrow \infty. \quad (48)$$

Then the query numbers (46) and (47) take asymptotic forms using (48)

$$S(K\tilde{K}) \sim \left\{ \frac{\pi}{4} + \left[\frac{\pi}{6} - \frac{\sqrt{3}}{2} + \frac{1}{5\sqrt{3}(2K\tilde{K})^2} \right] \frac{1}{\sqrt{K\tilde{K}}} \right\} \sqrt{N} \quad (49)$$

$$T(K, \tilde{K}) \sim \left\{ \frac{\pi}{4} + \left[\left(\frac{\pi}{3} - \frac{\sqrt{3}}{2} \right) - \frac{1}{4\sqrt{3}K} - \frac{19\sqrt{3}}{10(6K)^2} \right] \frac{1}{\sqrt{K}} + \left[\frac{\pi}{6} - \frac{\sqrt{3}}{2} + \frac{1}{5\sqrt{3}(2\tilde{K})^2} \right] \frac{1}{\sqrt{K\tilde{K}}} \right\} \sqrt{N} \quad (50)$$

As for the difference (55) of query numbers, the ratio K/\tilde{K} becomes relevant in determining the asymptotic behavior. There are 3 possibilities:

If $K/\tilde{K} \rightarrow 0$, then $1/K$ is dominating, and

$$T(K, \tilde{K}) - S(K\tilde{K}) \sim \left[\left(\frac{\pi}{3} - \frac{\sqrt{3}}{2} \right) K^{-\frac{1}{2}} \right] \frac{1}{\sqrt{N}}. \quad (51)$$

If $K/\tilde{K} \rightarrow \infty$, then $1/\tilde{K}$ is dominating, and

$$T(K, \tilde{K}) - S(K\tilde{K}) \sim \left(\frac{1}{20\sqrt{3}} K^{-\frac{1}{2}} \tilde{K}^{-\frac{5}{2}} \right) \frac{1}{\sqrt{N}}. \quad (52)$$

If $K/\tilde{K} \rightarrow \text{finite number}$, then we have the same result as (51). In both the expressions (51) and (52) the coefficients of $1/\sqrt{N}$ are positive. Up to now we saw that $T > S$. Now let us formally prove as a theorem (56) that $T > S$ in general, when $K \geq 2$ and $\tilde{K} \geq 2$.

B. General Proof that $T(K, \tilde{K}) > S(K\tilde{K})$

Now we prove that $T(K, \tilde{K}) - S(K\tilde{K})$ is always positive in the region $K, \tilde{K} \in [2, +\infty)$. In order to complete the proof we need the following two lemmas.

Lemma 1:

$$\frac{\pi}{4} + \left(\frac{1}{2}\alpha - \eta \right) (x) > 0, \quad \forall x \in [2, +\infty) \quad (53)$$

Proof: $\forall x \in [2, +\infty)$

The derivative $\left[\frac{\pi}{4} + \left(\frac{1}{2}\alpha - \eta \right) \right]' (x) = \frac{1}{4\sqrt{x}} f(x)$ with $f(x) \equiv \frac{3}{\sqrt{3x-4}} - \arctan \frac{\sqrt{3x-4}}{x-2}$. While $f'(x) = \frac{-9x+8}{2x(x-1)(3x-4)^{\frac{3}{2}}} < 0$, so that $f(x)$ monotonous decreasing. Further, since that $f(2) = \frac{3}{\sqrt{2}} - \frac{\pi}{2} > 0$, $f(x) \xrightarrow{x \rightarrow +\infty} 0$, then continuous function $f(x) > 0$ in the region. [$f(x)$ is positive at one point $x = 2$ and tends to zero at infinity. As a continuous and monotonous function, $f(x)$ can never become negative nor zero in the region.] Therefore $\left[\frac{\pi}{4} + \left(\frac{1}{2}\alpha - \eta \right) \right]' (x) > 0$, so that $\frac{\pi}{4} + \left(\frac{1}{2}\alpha - \eta \right) (x)$ is a monotonous increasing function of x . With $\frac{\pi}{4} + \left(\frac{1}{2}\alpha - \eta \right) (2) = \frac{3-2\sqrt{2}}{8}\pi > 0$, we conclude that $\frac{\pi}{4} + \left(\frac{1}{2}\alpha - \eta \right) (2) > 0$ in the region.

Lemma 2:

$$(\alpha - \eta) (x) \text{ monotonous decreasing,} \quad \forall x \in [2, +\infty) \quad (54)$$

Proof: $\forall x \in [2, +\infty)$

The derivative $(\alpha - \eta)' (x) = \frac{1}{4\sqrt{x}} g(x)$ with $g(x) \equiv \frac{\sqrt{3x-4}}{x-1} - \arctan \frac{\sqrt{3x-4}}{x-2}$. While $g'(x) =$

$\frac{1}{x(x-1)^2\sqrt{3x-4}} > 0$, so that $g(x)$ monotonous increasing. Further, since that $g(2) = \sqrt{2} - \frac{\pi}{2} < 0$, $g(x) \xrightarrow{x \rightarrow +\infty} 0$, then continuous function $g(x) < 0$ in the region. [$g(x)$ is negative at one point $x = 2$ and tends to zero at infinity. As a continuous and monotonous function, $g(x)$ can never become positive nor zero in the region.] Therefore $(\alpha - \eta)'(x) < 0$, we conclude that $(\alpha - \eta)(x)$ is a monotonous decreasing function of x in the region.

Having proved these two lemmas, we look at the structure of $T(K, \tilde{K}) - S(K\tilde{K})$ using (46) and (47):

$$\begin{aligned} & T(K, \tilde{K}) - S(K\tilde{K}) \\ &= \left\{ \left[\frac{\pi}{4} + \frac{1}{2}\alpha(K) - \eta(K) \right] \frac{1}{\sqrt{K}} + \left[(\alpha(\tilde{K}) - \eta(\tilde{K})) - (\alpha(K\tilde{K}) - \eta(K\tilde{K})) \right] \frac{1}{\sqrt{K\tilde{K}}} \right\} \sqrt{N}. \end{aligned} \quad (55)$$

Making use of **Lemma 1** (53), we see that the terms $\frac{\pi}{4} + \frac{1}{2}\alpha(K) - \eta(K)$ appearing in the first bracket of (55) is positive for $K \geq 2$. Making use of **Lemma 2** (54) and since $K\tilde{K} > \tilde{K}$, the monotony of $\alpha - \eta$ ensures that $(\alpha(\tilde{K}) - \eta(\tilde{K})) > (\alpha(K\tilde{K}) - \eta(K\tilde{K}))$. So that the second bracket of (55) is also positive for both $K \geq 2$ and $\tilde{K} \geq 2$. Therefore the whole brace of (55) is positive. As a consequence, we conclude our result in the following theorem

Theorem:

$$T(K, \tilde{K}) > S(K\tilde{K}), \quad \forall K, \tilde{K} \in [2, \infty). \quad (56)$$

i.e. Hierarchical partial search makes more queries to the oracle than direct partial search.
Direct GRK partial search works faster.

C. Hierarchy with Many GRK's

Theorem (56) can be extended to the case of hierarchical search with an arbitrary number of GRK's. The direct GRK always works faster. We prove the statement as follows.

Consider a hierarchy with m GRK's. Assume that $m \geq 2$. We denote the whole operations $G_1 G_2^{j_2} G_1^{j_1}$ of each GRK by one symbol and define an operator

$$\mathcal{G} \equiv G_1 G_2^{j_2} G_1^{j_1}. \quad (57)$$

The hierarchical search works on the initial state $|s_1\rangle$ as

$$\mathcal{G}_m \dots \mathcal{G}_3 \mathcal{G}_2 \mathcal{G}_1 |s_1\rangle, \quad (58)$$

where the sub-index denotes position of the GRK in the hierarchy [sequence]. The proof can be written formally in the following way. Define the total number of queries of the hierarchy

$$T(K_1, K_2, \dots, K_m) \equiv S(K_1) + \sum_{i=2}^m \bar{S}(K_{i-1}, K_i). \quad (59)$$

Here K_i is number of "sub"-blocks in the i^{th} partition of database. [We denoted K_1 and K_2 by K and \tilde{K} respectively in previous sections.] $S(K_1)$ is number of queries of the first GRK, and $\bar{S}(K_{i-1}, K_i)$ that of the i^{th} GRK in the hierarchy. Note that S and \bar{S} are not of the same function form. S takes the form corresponding to a direct GRK (25):

$$S(K_1) = \left(\frac{\pi}{4} + \frac{\alpha(K_1) - \eta(K_1)}{\sqrt{K_1}} \right) \sqrt{N}. \quad (60)$$

While \bar{S} takes a form of sequential GRK similar to (45):

$$\bar{S}(K_{i-1}, K_i) = \left(\frac{\pi}{4} - \frac{\alpha(K_{i-1})}{2} + \frac{\alpha(K_i) - \eta(K_i)}{\sqrt{K_i}} \right) \frac{\sqrt{N}}{\sqrt{\prod_{j=1}^{i-1} K_j}}, \quad i \geq 2. \quad (61)$$

[We denoted $S(K_1)$ and $\bar{S}(K_1, K_2)$ by $S(K)$ and $\bar{S}(K, \tilde{K})$ respectively in previous sections.]

Let us substitute these expressions into (59):

$$T(K_1, K_2, \dots, K_m) = \left\{ \frac{\pi}{4} + \sum_{i=1}^{m-1} \frac{\frac{\pi}{4} + \frac{1}{2}\alpha(K_i) - \eta(K_i)}{\sqrt{\prod_{j=1}^i K_j}} + \frac{\alpha(K_m) - \eta(K_m)}{\sqrt{\prod_{i=1}^m K_i}} \right\} \sqrt{N}. \quad (62)$$

On the other hand, if we partition the database directly into the smallest sub-blocks, then the number of these sub-blocks would be $\prod_{i=1}^m K_i$. A direct GRK will locate the smallest target sub-block. This would require

$$S\left(\prod_{i=1}^m K_i\right) = \left\{ \frac{\pi}{4} + \frac{\alpha(\prod_{i=1}^m K_i) - \eta(\prod_{i=1}^m K_i)}{\sqrt{\prod_{i=1}^m K_i}} \right\} \sqrt{N} \quad (63)$$

queries to the oracle. Therefore the difference of (62) and (63) is

$$\begin{aligned} & T(K_1, K_2, \dots, K_m) - S\left(\prod_{i=1}^m K_i\right) \\ &= \left\{ \left(\sum_{i=1}^{m-1} \frac{\frac{\pi}{4} + \frac{1}{2}\alpha(K_i) - \eta(K_i)}{\sqrt{\prod_{j=1}^i K_j}} \right) + \frac{[\alpha(K_m) - \eta(K_m)] - [\alpha(\prod_{i=1}^m K_i) - \eta(\prod_{i=1}^m K_i)]}{\sqrt{\prod_{i=1}^m K_i}} \right\} \sqrt{N}. \end{aligned} \quad (64)$$

We will show that this expression is always positive when each $K_i \geq 2$. Using **Lemma 1** (53), we see that each term under the summation of (64) is positive. Using **Lemma 2** (54), $\alpha - \eta$ is a monotonous decreasing function. Note that product of all K_i 's is larger than

K_m , we see that the remaining term in the brace of (64) is also positive. Consequently, we conclude our result in the following corollary.

Corollary:

$$T(K_1, K_2, \dots, K_m) > S\left(\prod_{i=1}^m K_i\right), \quad \forall K_i \in [2, +\infty). \quad (65)$$

i.e. Hierarchy of *arbitrary* number of GRK's makes more queries to the oracle than a direct GRK. *Direct GRK partial search always works faster.*

VII. SUMMARY

The present paper studied quantum search. Partial search algorithm is called GRK. We studied partial search hierarchy and compared it with direct partial search [GRK]. Consider database of N items with a single target item [target item also called marked item or solution]. The database is partitioned into K blocks, each block further partitioned into \tilde{K} sub-blocks. Hierarchical search is: use GRK and sequential GRK to find the target block and target sub-block, respectively. Successive GRK's can be made if the database is further partitioned. Each sequential GRK in the hierarchy works faster than the previous one. However, the total number of queries to the oracle adds up. The main conclusion is that *a partial search hierarchy works slower than a direct partial search*, see theorem (56) and corollary (65). For example, consider a database partitioned into 3 blocks. Each block is further partitioned into 3 sub-blocks, so totally there are 9 sub-blocks. One could first find the target block using GRK, then the target sub-block by a sequential GRK. Nevertheless, it is faster to run a GRK partial search directly over the 9 sub-blocks and finds the target sub-block once.

Note: Only the class of algorithms using the standard Grover oracle was considered in the paper. This means that if one has already built the main Grover algorithm experimentally, then we do not need any new hardware to run the GRK algorithm. Another advantage of using the same oracle I_t as the main Grover algorithm is more subtle: We can use ancilla [additional or auxiliary] q-bits to label different partitions of the database into blocks of equal size $b = N/K$. Then we are able to run GRK algorithm simultaneously for different partitions. [See Appendix D for more details.] Later a user can measure the ancilla q-bits and choose his or her favorite partition, by that time the target block already will be found.

Acknowledgments

The work is supported by NSF Grant DMS-0503712.

APPENDIX A:

DIFFERENCES OF THE LAST OPERATION OF GRK IN LITERATURES

Different versions of the last operation in **Step 3** of GRK appeared in literatures [6, 7, 9]. People have finalized [after steps 1 and 2] the state $|v\rangle \equiv G_2^{j_2} G_1^{j_1} |s_1\rangle$ with different operations I_{s_1} , $-I_t I_{s_1}$, or $G_1 \equiv -I_{s_1} I_t$. Grover and Radhakrishnan [6] used I_{s_1} . This makes one less query to the oracle but the amplitude of the target item is negative in the final state $I_{s_1}|v\rangle$. Paper [7] used $-I_t I_{s_1}$ but paper [10] used G_1 . The last two version become the same in the large block limit. This means that final states $-I_t I_{s_1}|v\rangle$ and $G_1|v\rangle$ are equivalent [of the same form] when $b \rightarrow \infty$, though I_{s_1} and I_t do not commute in general. We choose G_1 in our paper because it uses the same Grover iteration.

APPENDIX B:

RANGES OF PARAMETERS α AND η

We are going to specify ranges of parameters α and η introduced in (21). Because of the constraint (22) relating the two parameters, it is sufficient to specify the range of α . It was shown in [7] that amplitudes [of items in the database after GRK] depend on $\sin(2j_2\theta_2) \sim \sin(2\alpha)$ and $\cos(2j_2\theta_2) \sim \cos(2\alpha)$. So that it is sufficient to take values of α within one period: $\alpha \in [a, a + \pi]$, with a some real number determined later. We are looking for the exact boundaries of α set by physical considerations.

Query numbers (21) are non-negative:

$$j_1 = \left(\frac{\pi}{4} - \frac{\eta}{\sqrt{K}} \right) \sqrt{N} \geq 0, \quad (\text{B1})$$

$$j_2 = \frac{\alpha}{\sqrt{K}} \sqrt{N} \geq 0. \quad (\text{B2})$$

Total query number (23) should be less than that of a full Grover search:

$$j_1 + j_2 = \left(\frac{\pi}{4} + \frac{\alpha - \eta}{\sqrt{K}} \right) \sqrt{N} \leq \frac{\pi}{4} \sqrt{N}. \quad (\text{B3})$$

These three inequalities (B1), (B2) and (B3) yield that

$$0 \leq \alpha \leq \eta \leq \frac{\pi}{4}\sqrt{K}. \quad (\text{B4})$$

We use constraint (22) to express η as a function of α

$$\eta(\alpha) = \frac{1}{2}\sqrt{K}\text{Arctan}\left(\frac{2\sqrt{K}\sin 2\alpha}{K - 4\sin^2 \alpha}\right) \quad (\text{B5})$$

with function $\text{Arctan}(x)$ multi-valued. But according to (B4), we have

$$0 \leq \text{Arctan}\left(\frac{2\sqrt{K}\sin 2\alpha}{K - 4\sin^2 \alpha}\right) \leq \frac{\pi}{2}. \quad (\text{B6})$$

Therefore we could take the principal branch $\arctan(x)$. Now inequality (B4) becomes

$$0 \leq \alpha \leq \frac{1}{2}\sqrt{K}\arctan\left(\frac{2\sqrt{K}\sin 2\alpha}{K - 4\sin^2 \alpha}\right) \leq \frac{\pi}{4}\sqrt{K}. \quad (\text{B7})$$

This inequality determines range of α .

The solution of (B4) can be written in the following form:

$$0 \leq \alpha \leq \alpha_B(K). \quad (\text{B8})$$

Here the upper bound $\alpha_B(K)$ is a function of K . When $K = 2, 3$ or 4 , $\alpha_B(K)$ coincide with the singularities of $\eta(\alpha)$. [$K - 4\sin^2 \alpha = 0$ at these singularities.] When $K \geq 5$, values of $\alpha_B(K)$ can be solved numerically. As K increases, $\alpha_B(K)$ approaches a certain positive number $\alpha_B(\infty)$. This limit $\alpha_B(\infty) = 0.947747\dots$ is the solution of $\alpha = \sin(2\alpha)$. [Inequality $\alpha \leq \eta(\alpha)$ becomes $\alpha \leq \sin(2\alpha)$ as $K \rightarrow \infty$.] The value of $\alpha_B(K)$ always lies in between $\alpha_B(\infty)$ and $\frac{\pi}{2}$ when $K \geq 5$. We list these results in Table II.

TABLE II: Upper Bound of α

K	2	3	4	5	6	100	∞
$\alpha_B(K)$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	1.22683	1.15100	0.956221	0.947747

APPENDIX C:

MINIMIZATION OF THE TOTAL NUMBER OF QUERIES OF GRK

Here we give a proof that (24) is the global minimum of $\alpha - \eta$ under constraint (22). In Appendix B we used (22) to express η as a function of α

$$\eta(\alpha) = \frac{\sqrt{K}}{2}\arctan\left(\frac{2\sqrt{K}\sin 2\alpha}{K - 4\sin^2 \alpha}\right). \quad (\text{C1})$$

Now we define a function

$$f(\alpha) \equiv \alpha - \eta(\alpha) \quad (\text{C2})$$

which we want to minimize within the range $0 \leq \alpha \leq \alpha_B(K)$. We first prove that (24) is a local minimum of $f(\alpha)$.

1. Case $K \geq 3$

The first derivative of $f(\alpha)$ is

$$f'(\alpha) = \frac{16(K-1)\sin^4\alpha - 4K^2\sin^2\alpha + K^2}{16(K-1)\sin^4\alpha - 8K\sin^2\alpha - K^2}. \quad (\text{C3})$$

It vanishes at (24) with $\sin^2\alpha = \frac{K}{4(K-1)}$. We calculate next the second derivative

$$f''(\alpha) = \frac{4K\sin 2\alpha[4(K-1)(K-2)\cos^2 2\alpha + 16(K-1)\cos 2\alpha + (K-2)^2(K+2)]}{[16(K-1)\sin^4\alpha - 8K\sin^2\alpha - K^2]^2}. \quad (\text{C4})$$

Note that the value of the denominator at (24) is $\frac{K^6}{(K-1)^2}$, which is strictly positive as $K \geq 3$. The numerator is also positive because both $\sin 2\alpha$ and $\cos 2\alpha$ are positive at (24) with $K \geq 3$. [See [7] for the range of $\alpha(K)$.] Therefore $f'(\alpha) = 0$ and $f''(\alpha) > 0$ at the solution (24), so that (24) is a local minimum for $K \geq 3$.

2. Case $K = 2$

The case that $K = 2$ is more subtle. Expression (24) yields that $\alpha = \frac{\pi}{4}$ and $\eta = \frac{\pi}{2\sqrt{2}}$. However, both first (C3) and second (C4) derivatives of $\alpha - \eta(\alpha)$ vanish at this critical point. The third derivative is non-zero: $f'''(\alpha = \frac{\pi}{4}) = -4$. So we expand function $\alpha - \eta(\alpha)$ about the critical point

$$\alpha - \eta(\alpha)|_{K=2} = -\frac{4}{3!} \left(\alpha - \frac{\pi}{4} \right)^3 + \mathcal{O} \left(\left(\alpha - \frac{\pi}{4} \right)^4 \right). \quad (\text{C5})$$

We see that $\alpha = \frac{\pi}{4}$ is actually a saddle point due to the non-vanishing cubic term. The form (C5) suggests that if α goes greater than $\frac{\pi}{4}$, value of function $\alpha - \eta(\alpha)$ could be further reduced than the value at the saddle point. However, $\alpha = \frac{\pi}{4}$ is a boundary set by physical considerations [see Table II]. Definition of α and η in (21) involves query numbers j_1 and

j_2 , which are non-negative. Therefore $j_1 \equiv \left(\frac{\pi}{4} - \frac{\eta}{\sqrt{2}}\right) \sqrt{N} \geq 0$, i.e. $\eta \leq \frac{\pi}{2\sqrt{2}}$. Now we allow α to go beyond $\frac{\pi}{4}$ and write

$$\alpha = \frac{\pi}{4} + \delta, \quad \eta = \frac{\pi}{2\sqrt{2}} + \epsilon. \quad (\text{C6})$$

Here δ and ϵ are infinitesimals, $\delta > 0$. Then constraint (22) requires that

$$\epsilon = \delta. \quad (\text{C7})$$

So that η would be greater than the physically allowed maximal value $\frac{\pi}{2\sqrt{2}}$ and j_1 would be negative $j_1 = -\frac{\delta}{\sqrt{2}}\sqrt{N}$. This analysis showed that α can never go beyond $\frac{\pi}{4}$ and function $\alpha - \eta(\alpha)$ is minimized at this boundary. Therefore, expression (24) as a local minimum is also valid in the case that $K = 2$.

Now we have proved that the critical point (24)

$$\alpha(K) = \frac{1}{2} \arccos\left(\frac{K-2}{2(K-1)}\right) \quad (\text{C8})$$

is a local minimum of $f(\alpha)$. Note that $f(\alpha)$ is analytical as $0 \leq \alpha \leq \alpha_B(K)$ and there is no singularity in this range any more. Therefore we can show that this local minimum (C8) is also global by comparing the value of $f(\alpha)$ at (C8) with those at the boundaries. [We always have $0 < \alpha(K) \leq \alpha_B(K)$ and equality holds only for $K = 2$.] We list the comparison results for $K = 2, 3$ and 4 in Table III.

TABLE III: Comparison of values of $f(\alpha)$ at different points

K	$f(0)$	$f(\alpha(K))$	$f(\alpha_B(K))$
2	0	$\frac{\pi}{4} (1 - \sqrt{2}) \approx -0.325323$	$\frac{\pi}{4} (1 - \sqrt{2}) \approx -0.325323$
3	0	-0.337098	-0.313152
4	0	-0.339837	0

When $K \geq 5$, $f(0) = f(\alpha_B(K)) = 0$, while $f(\alpha(K)) < 0$. Therefore, we conclude that the critical point (24) or (C8) is always the global minimum.

APPENDIX D: DIFFERENT PARTITIONS OF A DATABASE

A data base of N items can be partitioned into blocks in different ways. For example, items in one block may have the first 3 bits of their addresses the same for one partition or

the last 3 bits the same for another partition. For a database partitioned into K blocks of equal size $b = N/K$, there are totally

$$P(N, K) = \frac{N!}{(b!)^K K!} \quad (\text{D1})$$

different ways of partition. We could use ancilla q-bits [also called additional or auxiliary q-bits] to label these partitions. As N and b both being large, we shall need

$$\log_2 P(N, K) \sim N \log_2 K - \log_2 K! \quad (\text{D2})$$

ancilla q-bits. For example, if we have $N = 4$ items and $K = 2$ blocks, then the number of partitions is $P(4, 2) = 3$ and we shall need $\log_2 3 \approx 2$ ancillas. In practice, The number (D2) can be further reduced if we only label the partitions commonly used, not all partitions. Then we can run GRK simultaneously for those selected partitions. When a user measures ancilla q-bit in his/her favorite partition, the target block will already be found by that time.

-
- [1] T. H. Cormen, C. E. Leiserson, R. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms* (MIT Press, 2003), 2nd ed.
 - [2] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge Univ. Press, 2000).
 - [3] R. Cleve, A. Ekert, L. Henderson, C. Macchiavello and M. Mosca, Complexity **4**, 33 (1998), quant-ph/9903061.
 - [4] L. K. Grover, *Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC)* (1996), 212, quant-ph/9605043.
 - [5] M. Heiligman, quant-ph/0006136.
 - [6] L. K. Grover and J. Radhakrishnan, *ACM Symp. on Parallel Algorithms and Architectures, Las Vegas, Nevada, USA* (2005), 186, quant-ph/0407122.
 - [7] V. E. Korepin, J. of Phys. A **38**, L731 (2005), quant-ph/0503238.
 - [8] V. E. Korepin and J. Liao, Quantum Information Processing **5**, 209 (2006), quant-ph/0510179.
 - [9] B.-S. Choi and V. E. Korepin, Quantum Information Processing **6**, 97 (2007), quant-ph/0608106.

- [10] V. E. Korepin and B. C. Vallilo, Prog. Theor. Phys. **116**, 783 (2006), quant-ph/0609205.
- [11] M. Mosca, Theoretical Computer Science **264**, 139 (2001).
- [12] G. Brassard, P. Hoyer, M. Mosca and A. Tapp, Contemporary mathematics **305**, 53 (2002).
- [13] V. E. Korepin and L. K. Grover, Quantum Information Processing **5**, 5 (2006), quant-ph/0504157.
- [14] Target item also called in literatures marked item or solution.
- [15] $(\sqrt{K}/2)(\pi/4)\sqrt{N}$ queries on average.